Levelset and B-spline deformable model techniques for image segmentation: a pragmatic comparative study.

Diane Lingrand, Johan Montagnat
Rainbow Team
I3S Laboratory UMR 6070
University of Nice - Sophia Antipolis / CNRS
930, route des Colles – B.P. 145 F06903 Sophia Antipolis Cedex- FRANCE
lingrand@essi.fr
http://www.i3s.unice.fr/~ lingrand

Abstract

Deformable contours are now widely used in image segmentation, using different models, criteria and numeric schemes. Some theoretical comparisons between few deformable model methods have been published [1]. Yet, very few experimental comparative study on real data have been reported. In this paper, we compare a levelset with a B-spline based deformable model approach to understand the mechanisms involved in these widely spread methods and to compare both evolution and results on different kind of image segmentation problems. In general, both methods yield to similar results. However, specific differences appear when considering particular problems.

1 Motivations

To model objects and segment images, both explicit [7, 18] and implicit [13, 2] deformable models have been proposed in the literature [12]. Among these methods, some focus on detecting edges characterized by high variation of features [10, 3], other on detecting regions characterized by homogeneity of spatially localized properties [17, 11]. Some other focus on both approaches.

The implicit deformable models are very commonly represented by levelsets. Levesets are widely used for 2D images segmentation [14, 3, 4] and 2D or 3D medical images segmentation [10, 15, 5, 6, 9, 8] among other areas. There exist many explicit model representations among which parametric models are the most widely used. In this paper, we consider the B-spline parametric models [16] as explicit representation.

Several studies on comparing different methods at a theoretical level has been published [1] but without concrete confrontation to real data. Our objective is to propose a comparative study of implicit and explicit deformable model based methods on concrete examples illustrating the differences between the two approaches. These methods are focused on the determination of a close contour of one or several objects. The initialization consists on a circle or another closed curve. This curve is iteratively modified according to an evolution (see figure 1):

$$\frac{\partial \Gamma}{\partial t} = F\mathbf{n}$$

where Γ represents the curve, $\mathbf{n}$ represents its normal, and F is computed from the image features and the intrinsic curve properties. In practice, we need a representation of the curve, an expression of the force F, and a numerical scheme to solve this equation.

2 Representation of a curve

We can distinguish three main methods of curve representation: polygonal, parametric and implicit. The polygonal representation is the simplest (see figure 2) but representing a smooth curves implies a model with a large number of points. A parametric model is defined as the set of points $P(t) = (x(t), y(t))$, where t is a real parameter. A lot of papers have been interested by different variety of splines because of there regularity properties (see figure 3). Finally, among implicit representation, there is a consensus about the levelset representation (see figure 4) which handles model topology changes in a simple and elegant manner.

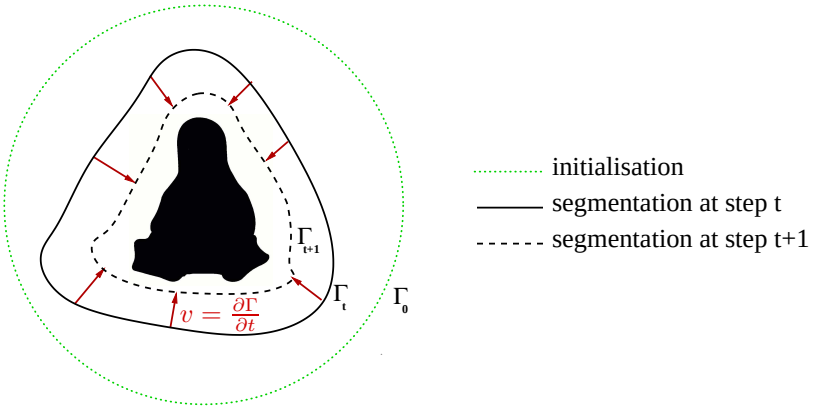


Fig. 1. Principle of image segmentation using deformable curves.

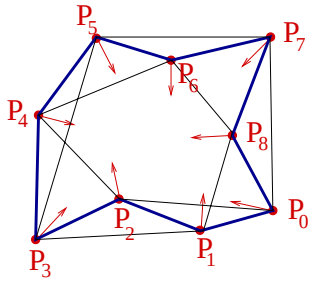


Fig. 2. The simplest representation of a curve: approximation by a polygon.

2.1 B-splines

A spline of degree n is defined as a piecewise polynomial of degree n. It has an order $(n-1)$ continuity. In order to simplify the parameterization of these splines, uniform B-Splines are considered.

A uniform B-Spline is defined by n ($n \geq 3$) control points Q_i and passes through points P_i defined by (see figure 3):

$$P_i = (Q_{i-1} + 4Q_i + Q_{i+1})/4 \tag{1}$$

Between two points P_i and P_{i+1}, a curve point $S_i(t)$ is defined by the parameter $t \in [0\ 1[$:

$$\begin{aligned} S_i(t) = [x(t)\ ,\ y(t)]^T = \quad & (-\tfrac{1}{6}Q_{i-1} + \tfrac{1}{2}Q_i - \tfrac{1}{2}Q_{i+1} + \tfrac{1}{6}Q_{i+2})t^3 + (\tfrac{1}{2}Q_{i-1} - Q_i + \tfrac{1}{2}Q_{i+1})t^2 \\ & +(-\tfrac{1}{2}Q_{i-1} + \tfrac{1}{2}Q_{i+1})t + \tfrac{1}{6}Q_{i-1} + \tfrac{2}{3}Q_i + \tfrac{1}{6}Q_{i+1} \end{aligned}$$

We can rewrite this as:

$$\begin{cases} x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \\ y(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 \end{cases}$$

with:

$$[a_0\ ,\ b_0]^T = (Q_{i-1} + 4Q_i + Q_{i-1})/6 \qquad [a_2\ ,\ b_2]^T = (Q_{i+1} - 2Q_i + Q_{i-1})/2$$
$$[a_1\ ,\ b_1]^T = (Q_{i+1} - Q_{i-1})/2 \qquad [a_3\ ,\ b_3]^T = (Q_{i+2} - 3Q_{i+1} + 3Q_i - Q_{i-1})/6$$

It is then easy to compute the normals and curvature using the first derivative of $x(t)$ and $y(t)$:

$$\mathbf{n} = \frac{1}{\sqrt{a_1^2 + b_1^2}} \begin{pmatrix} -b_1 \\ a_1 \end{pmatrix}, \kappa = 2(a_1 b_2 - a_2 b_1)(a_1^2 + b_1^2)^{\frac{3}{2}}$$

This representation is very light: only a limited number of points P_i are needed.

During model evolution, the force F is applied to points P_i. At each evolution step t, the corresponding control points Q_i are recomputed (by inversion of equation 1) in order to determine the B-spline curve (inside

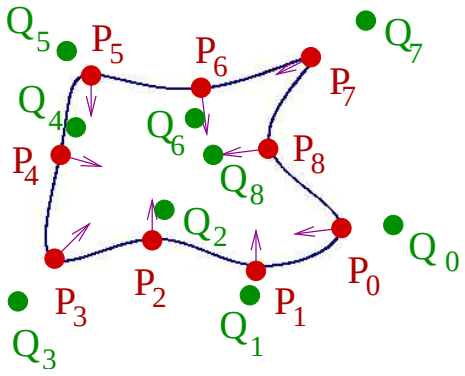


Fig. 3. Uniform B-spline defined by the control points Q_i and passing through the points P_i. Arrows represent normals.

and outside regions) and to estimate parameters such as normals and curvature. The computation of normals and curvatures is easy from the above equation. However, during evolution, several parts of the curve may self intersect. It is then needed to test if this occurs and to split the curve in two parts recursively when needed. Some other tests must be implemented in order to verify, after splitting operation, the orientation of the curve, or if two different curves converge toward the same region (one must be eliminated). However, when two curves intersect, there are different ways of dealing with it depending on the problem to solve:

- When a shape prior is introduced, and two objects overlap, one may want to segment both objects and preserve their intersection.
- One may want to obtain distinct but non intersecting regions with a common border.
- One may want to fuse the regions in a single one.

Last, the property of uniformity may also be altered by the evolution. It is then necessary to rearrange the points along the curve.

2.2 Levelsets

A 2D curve is defined by a 2D distance map which is an image where each pixel has a value corresponding to the distance between this pixel and the curve. This distance is signed assuming, for example, that points inside (resp. outside) the curve have negative (resp. positive) values. The curve is implicitly defined by the isolevel of value 0.

The normal and curvature are easy to compute on the curve:

$$\kappa = \text{div}\left(\frac{\nabla u}{|\nabla u|}\right) \text{ and } \vec{n} = \frac{\nabla u}{|\nabla u|}$$

where u represents the distance map. Normal and curvature can also be computed on a point outside the curve using the isolevel going through this point.

Contrarily to B-splines, the levelset representation implicitly handles problems of topology changes, intersection, superposition or orientation. However, some evolution criterion do not preserve the properties of the distance map: it must be reinitialized regularly.

The levelset representation is simple and easy to implement, but it requires a lot of memory and computations: the whole distance map storage and update is needed for this representation. However, some implementation tricks such as the narrow band may help to reduce the computation time.

3 Curve evolution

A deformable model evolution is steered by an energy minimization process: the solution of the segmentation is defined as the curve minimizing as an energy E which is depends on the image content (data term) and the curve

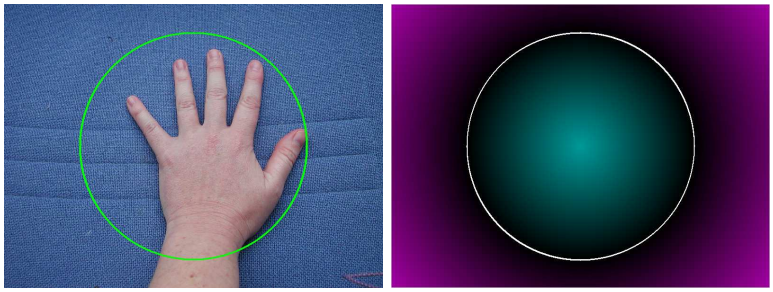


Fig. 4. On the left: original image with the isolevel 0 (initialization). On the right: distance map corresponding. Cyan colors (resp. magenta colors) corresponds to negative (resp. positive) distances.

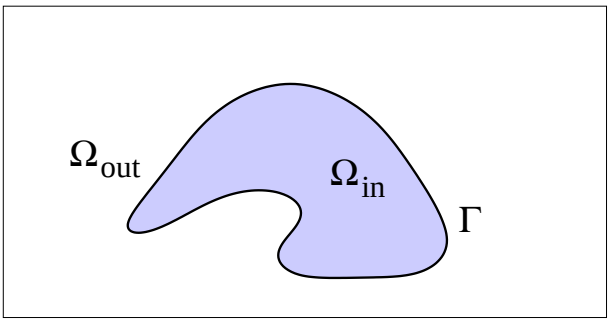


Fig. 5. Notations for the segmentation problem.

intrinsic regularity properties (internal term). The data term expresses the characterization of what is searched in an image. By segmentation, we mean the extraction of some objects boundaries from a background. Different characterizations of an object exist: using the the boundaries or the region inside the object. Boundaries are characterized by areas of large change of intensity or color. Regions are characterized by constant intensity of color over a uniformed background, range of colors, variance, texture, histogram...

In the following, we will consider the segmentation of a region Ω_{in} from a background Ω_{out} separated by a curve Γ (see figure 5).

Segmenting a uniform region from a uniform background. Assuming that we want to segment a uniform region Ω_{in} from a uniform background Ω_{out}, the energy is:

$$E = \int\int_{\Omega_{\text{in}}} (I(x,y) - \mu_{\text{in}})^2 dxdy + \int\int_{\Omega_{\text{out}}} (I(x,y) - \mu_{\text{out}})^2 dxdy$$

where μ_{in} (resp. μ_{out}) is the mean intensity of the region Ω_{in} (resp. Ω_{out}).

In order to minimize this energy E, we derive this expression with respect to the convergence step and obtain the force F:

$$F = \lambda_1 (I - \mu_{in})^2 - \lambda_2 (I - \mu_{out})^2$$

Many authors have observed that several curves may satisfy the evolution criterion because of the presence of noise in the images and the approximation made in order to model the curves. Among the possible solutions, we can decide to take the curve that minimizes the curve length:

$$E = \lambda_1 \int\int_{\Omega_{in}} (I(x,y) - \mu_{in})^2 dxdy + \lambda_2 \int\int_{\Omega_{out}} (I(x,y) - \mu_{out})^2 dxdy + \lambda_3 \int_{\Gamma} ds$$

Discretized:

$$E = \lambda_1 \sum\sum_{\Omega_{in}} (I(i,j) - \mu_{in})^2 + \lambda_2 \sum\sum_{\Omega_{out}} (I(i,j) - \mu_{out})^2 + \lambda_3 L_{\Gamma}$$

The force is then related to the curvature κ:

$$F = \lambda_1 (I - \mu_{in})^2 - \lambda_2 (I - \mu_{out})^2 + \lambda_3 \kappa \tag{2}$$

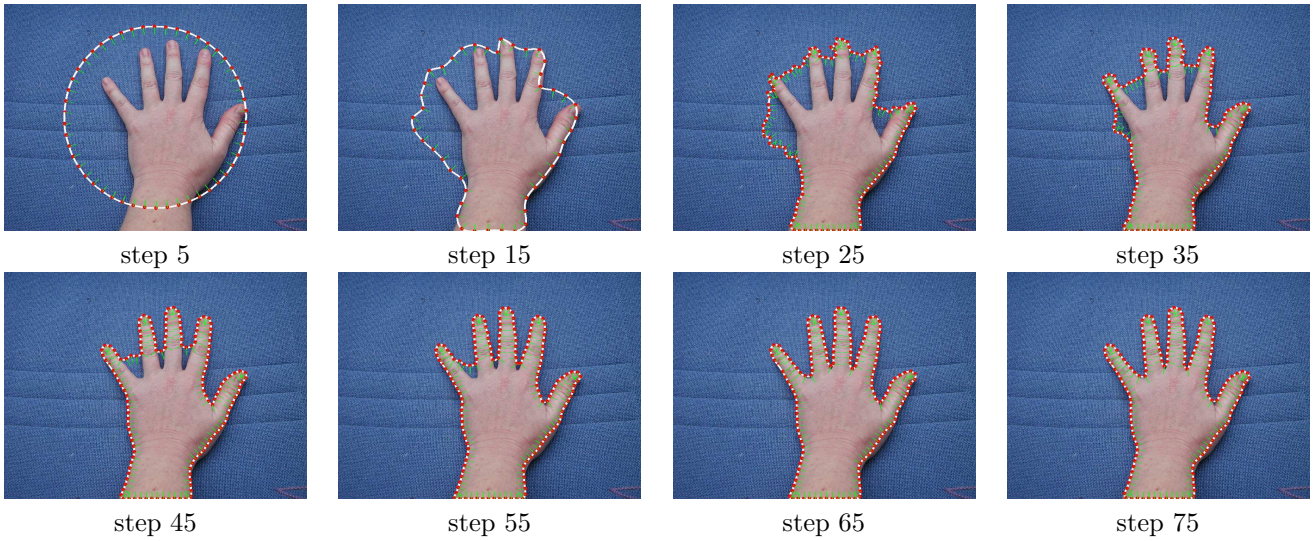


Fig. 6. B-spline representation. The convergence is reached after 75 iterations. The parameters used are $\lambda_1 = \lambda_2 = 0.001$ and $\lambda_3 = 1$ for iterations 1 to 15 and $\lambda_1 = \lambda_2 = 0.0005$ and $\lambda_3 = 1$ for iterations 16 to 60) and finally $\lambda_1 = \lambda_2 = 0.0001$ and $\lambda_3 = 1$. Points have been added in order to be able to represent the curve at iteration 15 (distance between points P_i limited from 30 to 50 then from 5 to 15 pixels). We observe that the convergence is difficult between the fingers.

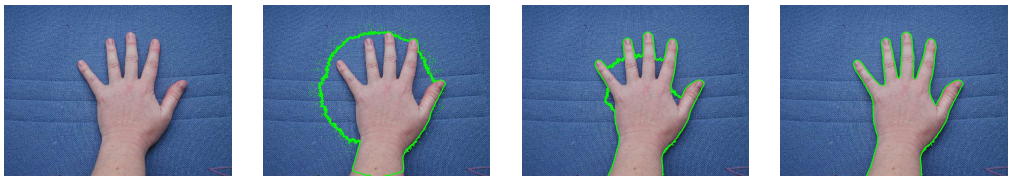


Fig. 7. Levelset representation. Convergence using 29 iterations with $\lambda_1 = 0.001$, $\lambda_2 = 0.001$ and $\lambda_3 = 1$.

4 Experimentations

Experiments in this section are done on real images taken with a digital camera. We have restricted the study to objects of uniform value on uniform background. Depending on the images, the value is the grey level, or a color component, or some function of a color component. The numeric scheme used in this paper is the simplest: constant iteration. This is not optimal but it is not the subject studied here.

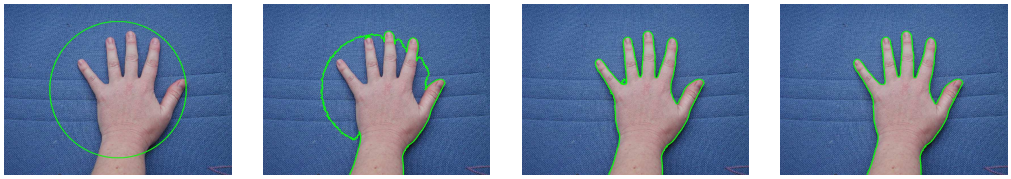
Experiments have been done using a software written in Java by the author. It is available under the GPL License at `http://www.i3s.unice.fr/ lingrand/ImageSegmentation.html`.

4.1 Segmenting objects of high curvature

In this experiment, we want to segment a hand from a uniform background. The image value used is the ratio of the red component from the sum of blue and green component. We use the evolution criterion (2) with both B-splines and levelset representation, each initialized by a circle centered on the image.

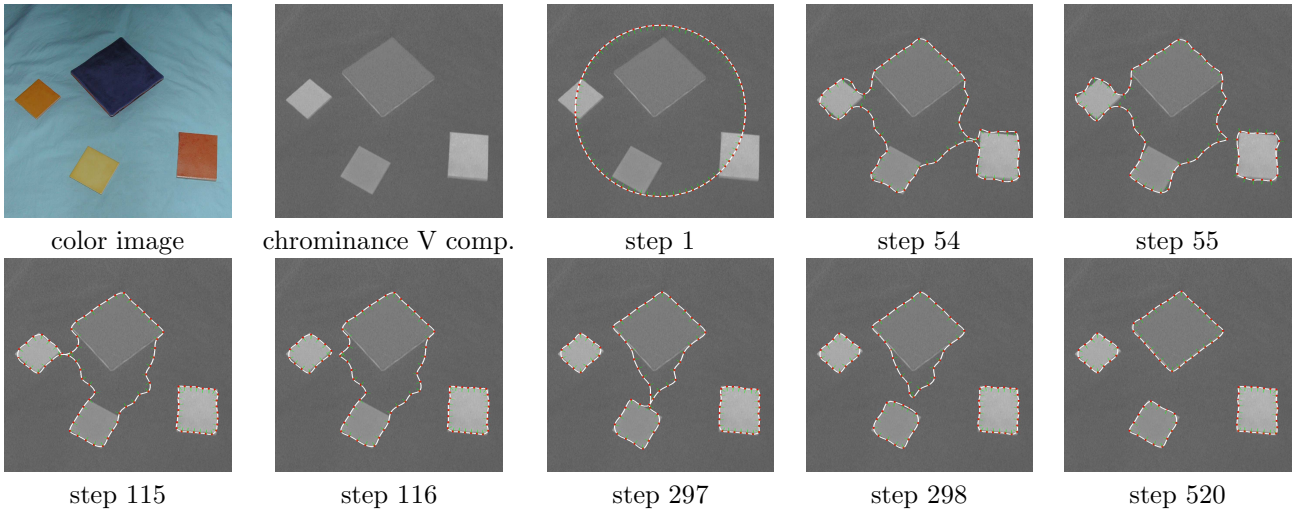
Using the B-splines representation, the curve converges easily to one shape but encounter difficulties to segment the fingers (see figure 6). We have helped the segmentation of the fingers by adding points to the model. It is not easy to automatically determine the number of points needed as it depends on the desired precision and curve smoothness.

Using the levelset method on the same image, we avoid the difficulties of area of high curvature. However, it is necessary to filter the image before the segmentation with a Gaussian filter (3x3) in order to lower the noise level. Figure 7 shows some artifacts that disappear using the filtering (see figure 8).



init step 5 step 10 step 11

Fig. 8. Levelset representation with Gaussian filter (8-connectivity). Convergence using 11 iterations with $\lambda_1 = 0.001$, $\lambda_2 = 0.001$ and $\lambda_3 = 1$.



color image chrominance V comp. step 1 step 54 step 55

step 115 step 116 step 297 step 298 step 520

Fig. 9. Chrominance component v . Using weights $\lambda_1 = 0.01$, $\lambda_2 = 0.01$, $\lambda_3 = 1$ and, after iteration number 50: $\lambda_1 = 0.001$, $\lambda_2 = 0.001$, $\lambda_3 = 1$. Distance between points P_i is limited from 30 to 50 pixels.

4.2 Segmenting several regions with different colors

In this experiment, we want to segment the different parts of figure 9 using the mean evolution criterion (2). We use the same initialization as previously: a circle centered on the image. Two problems arise: splitting the curve in several parts and having different regions of different intensities.

As seen before, the levelset representation intrinsicly handles the splitting. However, with the B-spline representation, it is necessary to test when the curve intersect itself and to split it in two curves.

Implementing the evolution criterion (2) using the levelset representation, μ_{in} is computed from points of negative distance map value while μ_{out} is computed from points of positive values. It is impossible to distinguish points inside one region from points inside another region: μ_{in} is common to all regions. Using the B-spline representation, we need to compute a mask of connected points for μ_{in} and μ_{out} computation: there is no difficulty to compute a new value μ_{in} for each region. Figures 9 and 10 show how the curve is split. At the beginning, both seems to converge well. But figure 10 shows that the levelset representation cannot deal with the darker region: the estimated mean is higher and this region has a mean intensity closer to the background. The algorithm does not segment this darker region.

The B-spline representation is more adapted to the segmentation of regions with different intensities. However, in figure 9, the squares are approximately segmented because of the small number of points. It would be necessary to add points to better represent corners. Another solution is to divide the image using the segmentation from the B-spline representation and to refine the result, locally, using levelsets.

5 Discussion and conclusion

Assuming that we want to segment objects of uniform intensity (or value) from a uniform background, both levelset and B-spline representation may fit except that (i) B-spline have difficulties to segment objects with high curvature and (ii) levelset are unable to distinguish one region to the others. However, depending on the

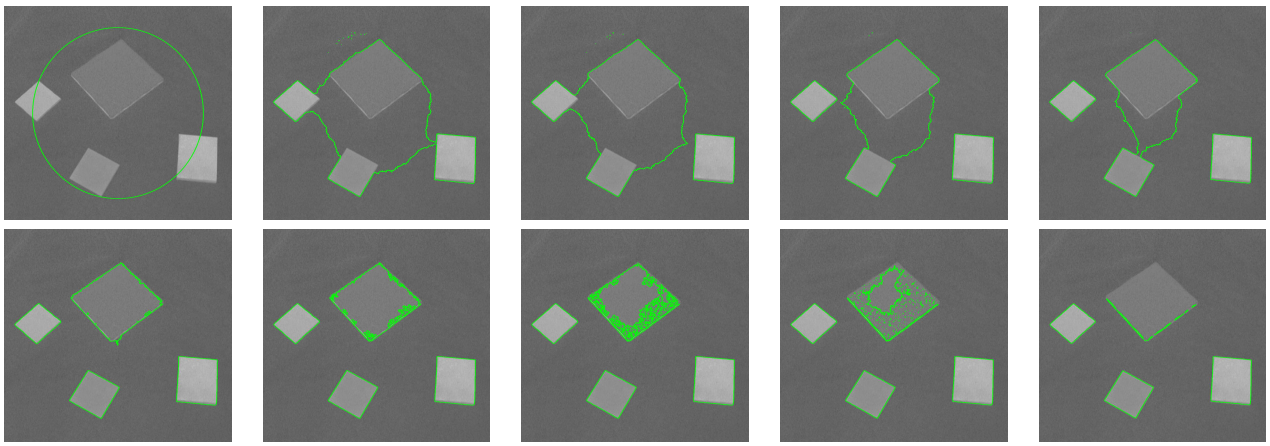


Fig. 10. Using Gaussian filter (3x3) and weights $\lambda_1 = 0.01$, $\lambda_2 = 0.01$, $\lambda_3 = 1$, convergence is achieved after 110 iterations.

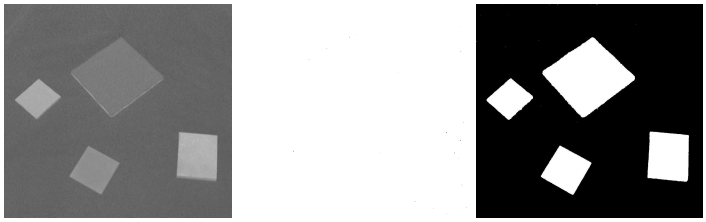


Fig. 11. Original image is first filtered using a Gaussian 3x3 filter and then thresholded (threshold value = 110). At last, small regions are deleted.

class of segmentation problem: high precision on one object, several objects to segment, same or different mean intensity, high curvature, necessity of a light representation, automatic or supervised method, we can make use of both representations.

Considering the problem of several different objects with high precision, we propose to first begin with the B-spline representation. The objects are well separated. Assuming that there is no occlusion in the scene, the image is separated in several parts and levelsets are introduced, using the result of B-spline segmentation for initialization. Is it the simplest and fastest method ? As seen in figure 11, there exists an older method that gives a satisfying result: very simple to implement, very fast. After a Gaussian filtering (3x3), the image is thresholded. Objects are well segmented with some small artifacts that are removed considering only regions of large area. This simplicity hide the problem of the choice of the threshold. Considering an application where this threshold can be calibrated (repeated segmentation in the same room with same luminosity ...), this is the method to be chosen. The PDE based methods do not need a threshold determination but are not fully automatic: the weights are important for the convergence.

As a conclusion, we can say that both methods are useful, separately or one helping another. The PDE based methods are more complex but avoid the problem or threshold determination. B-spline can handle different regions but the number of points must be chosen according to the precision needed. Levelset do not need to determine a number of points but cannot manage different regions if their mean intensity are different. Moreover, PDE methods take their importance for more complex problems using more complex regions properties.

References

[1] Gilles Aubert and Laure Blanc-Féraud. Some remarks on the equivalence between 2D and 3D classical snakes and geodesic active contours. *International Journal of Computer Vision*, 34(1):5–17, September 1999.

[2] Vicent Caselles, Francine Catte, Tomeu Coll, and Françoise Dibos. A geometric model for active contours in image processing. In *Numerische Mathematik*, volume 66, pages 1–33, 1993.

[3] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.

[4] L.D. Cohen and Ron Kimmel. Global minimum for active contour models: A minimal path approach. *Int. J. of Computer Vision*, 24(1):57–78, 1997.

[5] M. Droske, B. Meyer, M. Rumpf, and C. Schaller. An adaptive level set method for medical image segmentation. In R. Leahy and M. Insana, editors, *Proc. of the Annual Symposium on Information Processing in Medical Imaging*. Springer, Lecture Notes Computer Science, 2001.

[6] José Gomes and Olivier Faugeras. Segmentation of the inner and outer surfaces of the cortex in man and monkey: an approach based on Partial Differential Equations. In *Proc. of the 5th Int. Conf. on Functional Mapping of the Human Brain*, 1999.

[7] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. SNAKES: Active contour models. *International Journal of Computer Vision*, 1:321–332, January 1988.

[8] Diane Lingrand, Arnaud Charnoz, Pierre Malick Koulibaly, Jacques Darcourt, and Johan Montagnat. Toward accurate segmentation of the LV myocardium and chamber for volumes estimation in gated SPECT sequences. In T. Pajdla and J. Matas, editors, *8th ECCV*, volume LNCS 3024, pages 267–278. Springer-Verlag, May 2004.

[9] Ravi Malladi and James A. Sethian. A Real-Time Algorithm for Medical Shape Recovery. In *International Conference on Computer Vision (ICCV'98)*, pages 304–310, Bombay, India, January 1998.

[10] Ravi Malladi, James A. Sethian, and Baba C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, February 1995.

[11] Tim McInerney and Demetri Terzopoulos. Deformable models in medical image analysis : a survey. *Medical Image Analysis*, 1(2):73–91, 1996.

[12] Johan Montagnat and Hervé Delingette. A review of deformable surfaces: topology, geometry and deformation. *Image and Vision Comput.*, 19(14):1023–1040, Dec. 2001.

[13] Stanley Osher and James A. Sethian. Fronts propagating with curvature dependent speed : algorithms based on the Hamilton-Jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.

[14] Nikos Paragios and Rachid Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):266–280, March 2000.

[15] Nikos Paragios, Mikael Rousson, and Visvanathan Ramesh. Knowledge-based registration and segmentation of the left ventricle: A level set approach. In *IEEE Workshop on Applications in Computer Vision, Orlando, Florida*, December 2002.

[16] Frédéric Precioso and Michel Barlaud. B-spline active contour with handling of topological changes for fast video segmentation. *EURASIP Journal on Applied Signal Processing, Special issue on Image Analysis for Multimedia Interactive Services - Part II, vol.2002 (6)*, pages 555–560, June 2002.

[17] R. Ronfard. Region-based strategies for active contour models. *International Journal of Computer Vision*, 13(2):229–251, 1994.

[18] Demetri Terzopoulos, Andrew Witkin, and Michael Kass. Constraints on deformable models : Recovering 3d shape and non rigid motion. *Artificial Intelligence*, 36(1):91–123, 1988.